Measures capacity up to 9999 mAh and samples the voltage

Battery Discharger/Capacity Meter

Part 1: design and control module

Design by B. Stuurman

This circuit can measure the capacity of individual cells as well as battery packs with an open circuit voltage of up to 35 V. The only requirement is that the discharged cell voltage has to be between 0.5 V and 2.5 V. During the discharge period, the instantaneous voltage is measured at one minute intervals and stored in memory. This allows the voltage curve to be drawn once the discharge process is completed.

As can be seen from the title photograph, the appearance of the battery discharger is simple and clearly arranged. There is a small keyboard with 12 keys for the input of data. All information is displayed on a 16-character LCD. The battery is connected to the two instrument terminals, a black one and a red one. In addition there are two slide switches, one is for on/off, the other one we will describe later. On the back is a sizeable heatsink onto which the discharge transistor is mounted. There are also eight large holes in the front panel, because the enclosure contains a cooling fan that blows air over the heatsink during the discharge period. This allows the power dissipation capacity to be nearly doubled.

The discharger is perfectly serviceable as a stand-alone unit, but as will become clear later on, a PC will be indispensable in certain situations. For this purpose, on the left side of the enclosure, there is a nine-way connector, which may be connected to the serial port of a PC. A diskette with the pertinent software will be available from *Elektor Electronics* Readers' Services.

Operation

Once the device as been switched on, the text 'NiCad Discharger' appears on the LCD. Pressing the '*'-key causes option menu 1 to be displayed, pressing the key again results in option menu 2, etc. There are six option menus in total. The '#'-key is the confirmation button, which may be compared (functionally) with the Enter-key on a PC-keyboard. Once the desired option menu has been reached by pressing the '*'-key one or several times, pressing the '#' activates the option. If this is a menu where data can be entered and the '#'-key remains depressed, then the current data is displayed on the LCD. When the '#'-key is released, question marks appear and new data can be entered using the number keys on the keyboard. The entry has to be confirmed by pressing the '#'-key.

While operating the keyboard, a time-out routine runs in the background. If there has been an entry error, or when just viewing the current value, the message 'NiCad Discharger' will reappear on the LCD after a few seconds.

The menus

There are six option menus. We will briefly describe them because in one fell swoop all the possibilities (and impossibilities) of the discharger will become immediately clear. The purpose of the 'Set'-menus is for the entry of data and behave nearly identically, the other menus are switching menus.

Set loadcurrent? If the '#'-key is pressed and held the currently set discharge current is displayed on the display. When the key is released the message 'Current: ???? mA' appears and a new value can be entered. If you do not want to change the value, you can either wait a while or press the '#'-key (as

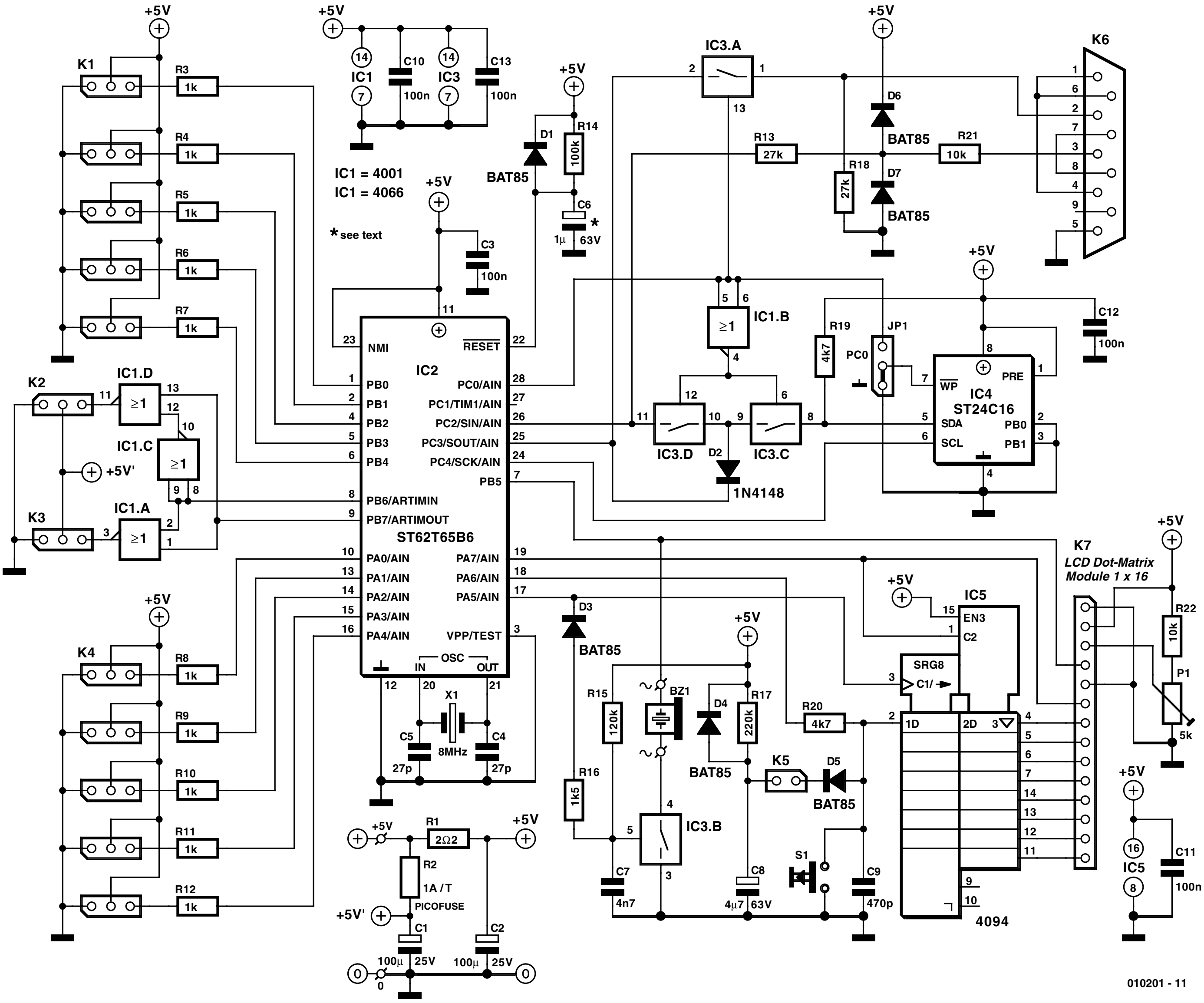


Figure 1. The 'heart' of the control module is formed by a microcontroller of the type ST62X65.

COMPONENTS LIST

Control board

Resistors:
R1 = 2O2
R2 = fuse, 1AT (time lag) Picofuse (e.g., Farnell 446-944)
R3-R12 = 1kΩ
R13,R18 = 27kΩ
R14 = 100kΩ *
R15 = 120kΩ
R16 = 1kΩ5
R17 = 220kΩ
R19,R20 = 4kΩ7
R21,R22 = 10kΩ
P1 = 5kΩ preset

Capacitors:
C1,C2 = 100µF 25V radial
C3,C10-C13 = 100nF ceramic
C4,C5 = 27pF
C6 = 1µF 63V radial*
C7 = 4nF7, lead pitch 5mm
C8 = 4µF7 63V radial
C9 = 470pF

Semiconductors:
D1*,D3-D7 = BAT85
D2 = 1N4148
IC1 = 4001
IC2 = ST62T65, programmed, order code **010201-41** (see Readers Services page)
IC3 = 4066
IC4 = ST24C16
IC5 = 4094

Miscellaneous:
K1-K4 = 12 x 3-way pinheader
K5 = 2-way pinheader
K6 = 9-way sub-D socket female), angled pins, PCB mount
K7 = 14-way SIL-header
JP1 = 3-way pinheader with jumper
S1 = pushbutton type D6 (ITT/Schadow)
X1 = 8MHz quartz crystal
Bz1 = AC buzzer
LCD dot matrix module, 1x16 characters, 80x36x10 mm (e.g., Farnell # 142-542)
14 way female PCB connector for LCD PCB, order code **010201-1** (see Readers Services page), supplied with discharger board **010201-2**)

**) not fitted when using ST62T65 type C*

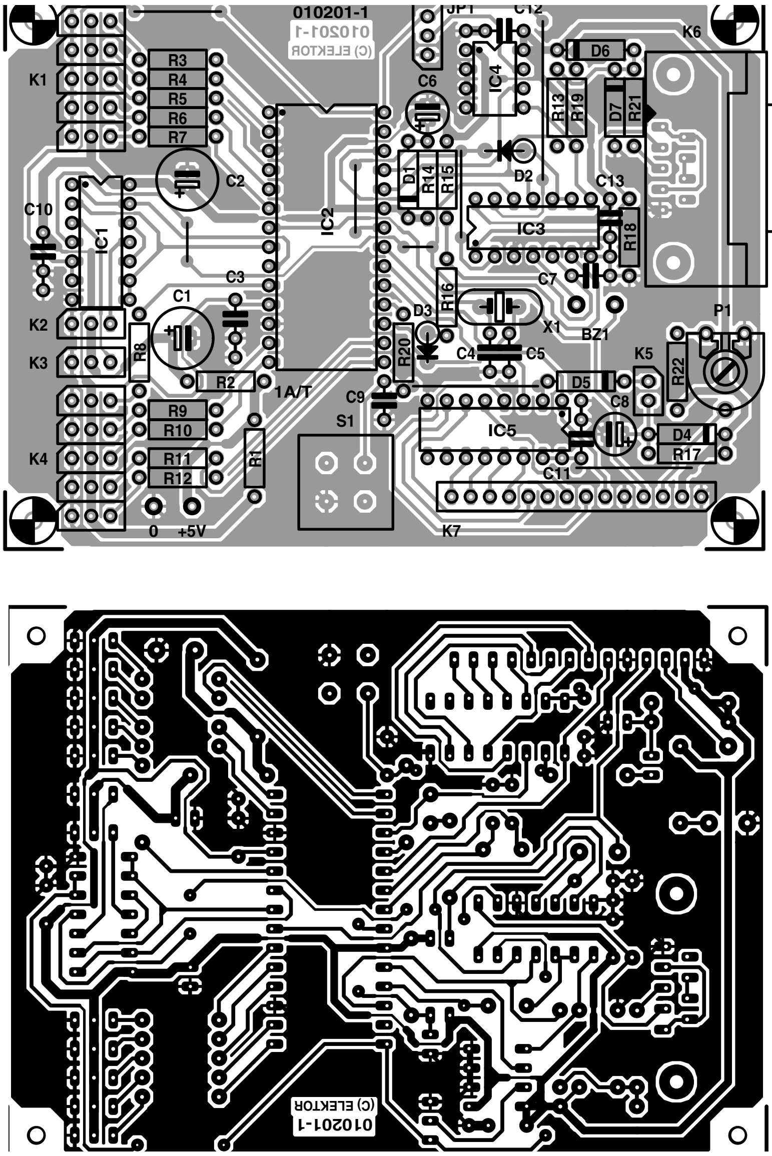


Figure 2. Copper track layout and component overlay of the control PCB.

confirmation) the message '! not accepted !' will appear on the display, followed almost immediately with the return to the option menu. The minimum current is 100 mA, the maximum is 2500 mA. The adjustment is in steps of 10 mA. Values less than 5 are rounded down and values greater than 5 are rounded up.

Set nr of cells? The number of cells in the battery pack can be entered here (maximum of 20).

Set volts/cell? The purpose of this option is to adjust the voltage at which a cell is considered discharged. The value must be in the range 0.5 - 2.5 V. For NiCd- or NiMH-cells which are being discharged at a current smaller than 0.5 C, 1.1 V is an appropriate switch-off value. At higher discharge currents a value of 1 V or even 0.9 V may be selected to counteract the voltage drop as a consequence of the internal resistance. Battery holders with spring contacts can have considerable contact resistance and can cause adverse effects! The wide voltage range allows lead-acid batteries to be discharged as well.

Start discharge? After confirmation with the '#'-key the voltage of the battery is compared with the shut-off value (= number of cells x voltage/cell). If it is higher, the discharge process commences. First, the sampling memory is cleared ('Clearing memory!'), then the current is raised to the selected value ('Setting Current!'). When this value has been reached, the cooling fan is turned on and the time is reset to zero. During the discharge the elapsed time and discharged capacity appear alternately on the display. The option menus remain active during the discharge process, with the understanding that the values can be displayed but cannot be altered. The

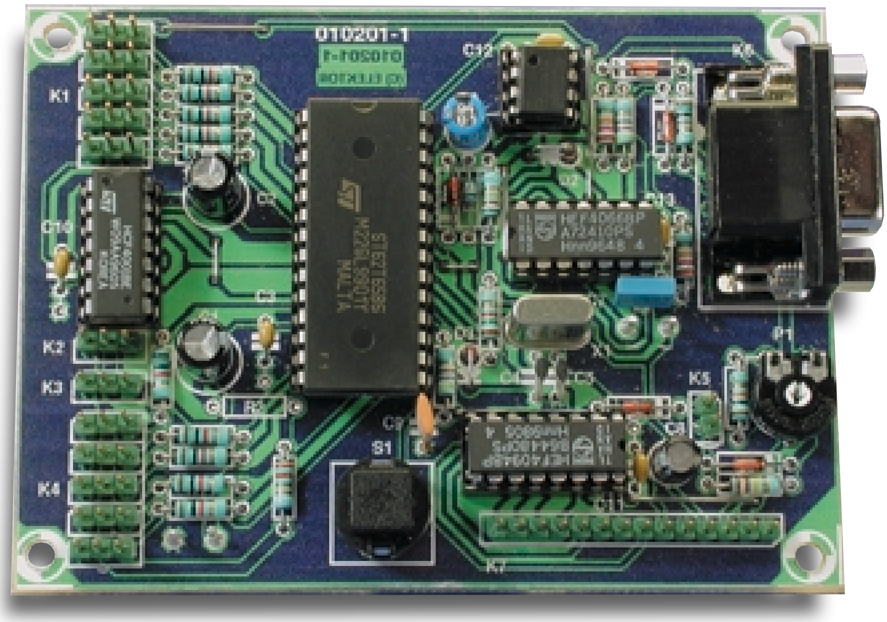


Figure 3. Photo of the completely assembled control board.

discharge process continues until the set voltage is reached or...

Stop discharge? This allows the discharge process to be terminated prematurely.

See last result? Once the discharge process has been completed, the discharge time and measured capacity are alternately shown on the LCD and an audible signal is given. This information disappears when any key is pressed. This option menu allows the measured results to be displayed again.

Plan of attack

A two-stage approach has been chosen. For starters, a small control module has been built which has acquired the name 'Chip' and contains all the principal hard- and software components. The 'heart' is formed by a programmed microcontroller. This microcontroller does not contain the software to control the discharger, but instead contains an 'interpreter' which reads and executes programs in a higher programming language. The 'higher programs' are stored in an EEPROM and can be directly downloaded from a PC.

The program for the discharger is written in the 'Chip'-language and built with a 'Chip'-assembler.

Control Module

Figure 1 shows the schematic for the control module 'Chip'. A central position is occupied by the microcontroller ST62T65 from ST-Microelectronics (IC2). The microcontroller contains a number of extra components, such as a standard timer, an auto-reload timer, an analogue/digital converter, and a serial interface, abbreviated to 'SPI'. The standard timer is used as a clock and keeps track of seconds, minutes, hours, day of the week and weeks in the year. Various software timers are also controlled by the standard timer. The auto-reload timer can be used, among other things, to control two servos independently, but can also be reprogrammed to generate a voltage. The fine adjustment of the clock is achieved with a software-byte and can be performed by the user. This value is loaded in the timer once a minute.

Inputs and outputs

Five digital outputs (Out0 -Out4) are available on K1, fitted with series resistors to protect the microcontroller. These outputs can be set or cleared and the state of each output can be read back. The servo outputs K2 and K3 are for use with regular servos or a voltage can be obtained here. Terminal block K4 makes five inputs available (Input0 - Input4), which can be read in either digital or analogue mode.

Sub-D connector K6 allows communication with a PC. The input signal is clamped by R21, D6 and D7 and, via R13, applied to the input of the SPI (Sin). The serial output signal (Sout) reaches the output pin of the connector via analogue switch IC3A. When the analogue switch is open, R18 pulls the output low. The signal levels do not comply with RS232 standards, but this circuit works reliably nonetheless, provided the connecting cable is not too long. During serial communications, PC0 is high and IC3A is closed, while IC3C and IC3D are open. By pulling PC0 low, EEPROM (IC4) is connected to the SPI and write enabled via WC. Software routines can now access the EEPRROM for read and write operations using the I^2C-protocol. The actual data transfer is handled by the SPI with a clock frequency of 308 kHz.

LC display

Connector K7 allows the connection of a one-line 16-character LCD. A serial to parallel converter (IC5) is used to drive the data bus of the LCD, so that only three outputs from the microcontroller are necessary for communication. PA5 provides the clock, PA6 the data and PA7 is the strobe. The strobe for IC5 is positive so that PA7 can also be used as (negative) strobe to clock the data into the LCD. PA6 is normally an input. Pushbutton S1 is connected to this input. This button is tested once per second by the operating system and if it is activated the currently running Chip program is terminated and control returns to the command processor. S1 is therefore not a reset button, but functions as a Break button — the real-time clock continuous to run. The connection Reg/Sel on the LCD serves the purpose of selecting between character mode or command mode. This terminal is driven by PB5. PB5 is also used to control buzzer BZ1. At the first falling edge of the clock — to transfer data for the LCD into the shift register — analogue switch IC3B is opened and the buzzer is switched off (D3, R15, C7). After the data transfer is finished, C7 will charge via R15 and IC3B will close. The contrast of the LCD can be adjusted with P1.

A few loose ends

Power-up reset is provided by D1, R14 and C6. It is possible to make the Chip interpreter self-starting by placing a jumper on K5. This way, C8 and D5 'hold down' S1 for a moment when Chip is switched on. R17 charges C8 and causes the pushbutton to be released. If a jumper is not connected to K5, the same result can be obtained by holding down S1 while switching on.

A final remark about the schematic. The part number shown for the controller has the suffix 'B'. The newer version ST62X65C may be used instead. In this case, if the LVD-option (low voltage detection) of this microcontroller is enabled, D1, R14, and C6 may be omitted. So take note of which type you are using.

Construction

Figure 2 shows the (single sided) printed circuit board for the control module and the corresponding component overlay. Incidentally, this PCB is delivered in combination with the discharger module that will be described in Part 2.

First mount the eight wire links, not forgetting the one under the socket for IC2. This is followed by the low profile components and finally it is the turn of the tall components, such as electrolytics C1 and C2. Only fit the ICs in their sockets once the entire PCB has been built and has been carefully checked.

If a different EEPROM (IC4) than the indicated type ST24C16 is used then this has implications for pin 7. The function of WC may be different with other EEPROMs and this may cause the write operation to be blocked. If this is the case, pin 7 may be connected to ground using jumper JP1.

The connection to the LCD may be made with a ribbon cable or a set of individual flexible wires with a length of 15 cm. A 14-pin female PCB header is used to connect to K7. Initially, a 4.5 V flat battery (3LR12) will suffice for the power supply (4.5 to 6 V).

Figure 3 shows a photo of the completed circuit board for control module. It wouldn't be a bad idea to carefully compare your own specimen with this.

The operating system

The control module communicates via the serial interface with the terminal program running on the PC. The interface parameters are set to **19200,7,n,2**. The project software contains the terminal program `CHIPTERM.EXE` (and the Windows-version '`VBTERM`') which can be used to download the programs. The `README.TXT` file contains the operating instructions.

Only lower-case letters may be used for input. After entry of the command and any parameters, if appropriate, the line is terminated with an 'Enter' after which the command processor jumps into action. Mistakes can be corrected with 'Backspace' and the entire line can be discarded with 'Escape'.

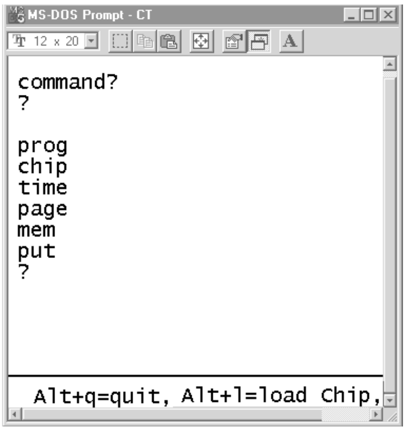
There are seven commands; we will deal with them briefly because they are essentially self-explanatory. When the control module is switched on and the terminal program has been started, pressing Enter will result in the display of 'command?' and on the next line the prompt '?'. By entering the command '?' (terminated with Enter) the entire command set is displayed on the screen (refer **Figure 4**):

prog [address]: this command allows reading and writing of the EEPROM. Only the even addresses are displayed, followed by the bytes that are stored on this and the successive address. Every Chip-instruction comprises two bytes. The lowest address is 000h and the highest is 7FFh. Using the '+' and '–' keys the address can be increased or decreased. When two bytes have been entered they are stored and the address is incremented. Escape is used to return to the command processor.

chip: this command starts the program, written in Chip-language, at address 000h. At this point the command processor ceases to run. A Chip-program can also be started by holding down S1 during power-up or by fitting a jumper on K5. The program can be interrupted by pushing S1. The break instruction or an error in the program will also cause termination of the program.

time: after entering this command a (running) clock appears on the screen. Pressing 'Enter' brings up the week field, the clock can be adjusted. It is a 24-hour clock. Pressing 'Escape' twice leaves the clock unaltered.

page, **mem** and **put:** these commands are used to read and write the registers in the microcontroller. We first have to make a few remarks about the address range 00h-3Fh. The microcontroller has, in addition to the normal RAM, two EEPROM-pages and an extra RAM-page. Using the page command one of these pages is mapped into the address range 00h-3Fh. In EEPROM-page 0 in address range 00h-0Fh we place a start-up message and in address 10h the adjustment byte for the clock. First select with 'mem 0' address 00h of the microcontroller. This address (in the extra RAM-page) and corresponding byte appear on the screen. With 'page 0' we select the EEPROM page 0. With the command 'put byte', 'byte' can be written to this address. Pressing the + key increments the address (the – key decrements). The start-up message can now be entered into the (microcontroller) EEPROM:



```
command?
?

prog
chip
time
page
mem
put
?

 Alt+q=quit, Alt+l=load Chip,
```

Figure 4. The command set as it appears on the screen.

2a, 20, 48, 69, 2c, 20, 49, 27, 6d, 20, 43, 68, 69, 70, 20, 2a

After all bytes have been entered into the EEPROM we press 'Escape' to return to the command processor. If you have a list of ASCII-characters then you will probably already know what the start-up message is. To display this message on the LCD it is necessary to reset the control module by briefly turning the power off and then turning it back on. At address 10h of the same page we place, in exactly the same manner, the adjustment byte for the clock. The clock will probably keep proper time with a value of EAh. Therefore: 'mem 10', 'page 0', 'put ea'. Increasing this value causes the clock to run slower, reducing it will make it run faster. The effect will only be noticeable after 24 hours or more. Don't forget to reset the control module, because this byte is only read when the unit starts up, just like the start-up message.

(010201-1)

In Part 2 we continue with the keyboard, the construction and installing the NiCad discharge program.